

DESENVOLVIMENTO DE UMA ESTAÇÃO ROBÓTICA EDUCACIONAL

Pedro K. Paim – pedropaim@terra.com.br
Lélio R. Soares Jr. – lelio@unb.br
Víctor H. Casanova Alcalde – casanova@unb.br
Laboratório de Controle e Visão por Computador
Grupo de Instrumentação, Controle e Automação (GICA)
Departamento de Engenharia Elétrica
Universidade de Brasília
70.910-900 - Brasília, DF

Resumo: *O trabalho descreve o desenvolvimento de uma estação robótica educacional como plataforma para experiências de laboratório em cinemática, geração de trajetórias, execução de tarefas, dinâmica e controle de robôs. A estação será usada nas aulas práticas das disciplinas de robótica nos cursos de engenharia elétrica, engenharia de controle e automação, etc. A estação foi desenvolvida em torno do robô Rhino XR4, um manipulador robótico de 5 eixos acionados por servomotores DC que incorporam engrenagens e codificadores óptico digitais. Formulou-se um modelo cinemático para a solução da cinemática direta e inversa, cujas soluções foram implementadas no ambiente Matlab. Para visualização da cinemática do robô desenvolveu-se um programa de simulação gráfica tri-dimensional, que usa uma estrutura wire-frame do robô e funções de interface gráfica do Matlab. Como adquiriu-se unicamente a estrutura mecânica do robô, projetou-se e construiu-se um controlador multi-axial e desenvolveu-se uma interface de programação para sensoreamento, processamento e acionamento do manipulador. O controlador multiaxial utiliza placas de controle de motores baseadas em microcontroladores, um conversor de protocolo de comunicação serial e um computador PC. Com o controlador em funcionamento pode-se verificar as resoluções de junta e a envelope de trabalho no espaço de juntas. A programação do sistema inclui interface de usuário, diagnóstico, configuração e controle do robô. O controlador foi posteriormente aplicado na execução de trajetórias e na realização de tarefas.*

Palavras Chave: *Ensino em controle e robótica, Robótica, Cinemática de robôs, Controle de robôs, Simulação de robôs.*

1. INTRODUÇÃO

Para complementar o ensino e pesquisa na cinemática, planejamento de trajetórias, dinâmica e controle de sistemas robóticos, iniciou-se o desenvolvimento de uma estação robótica educacional. Como somente adquiriu-se a estrutura mecânica do robô Rhino XR4, que inclui os motores de acionamento das juntas, houve necessidade de projetar e construir o controlador, as interfaces de comunicação, o ambiente de programação. A estação robótica em vista consiste de um robô, o controlador, um sistema de visão por computador e acessórios, compondo uma célula de trabalho. Na versão inicial (RORIZ, 2000) desenvolveu-se uma interface de programação no ambiente C Builder e algumas funções no ambiente Matlab. Posteriormente, integrou-se toda a operação e controle do robô no ambiente Matlab,

para o qual desenvolveram-se funções específicas (GUIMARÃES, 2002). Neste trabalho consolidam-se as operações e desenvolvem-se as experiências de manipulação robótica.

Não dispondo de informações detalhadas do fabricante sobre o controlador e conhecendo as limitações na programação do mesmo (MRAD e DEEB, 1997), decidiu-se por uma arquitetura inovadora e modular. Na seção 2 descreve-se o robô educacional Rhino XR4 e algumas informações sobre seu controlador. A solução analítica dos problemas de cinemática direta e inversa do robô e sua implementação no ambiente Matlab são tratados na seção 3. Na seção 4 apresenta-se o desenvolvimento de um programa de simulação gráfica tri-dimensional do robô Rhino XR4, resultado da integração da simulação da cinemática com os recursos de interface gráfica do Matlab. A seção 5 trata do controlador multiaxial projetado e construído e da sua aplicação na verificação das resoluções de junta e da geometria do volume de trabalho no espaço juntas. A seção 6 descreve o desenvolvimento da programação desenvolvida para diagnóstico, configuração e controle do robô e sua posterior aplicação no acompanhamento de trajetórias e realização de tarefas.

2. ROBÔ RHINO XR4

O robô Rhino XR4 é mostrado na Figura 1. Trata-se de um robô educacional de mesa com 5 eixos de coordenadas articuladas, com acionamento elétrico e controle independente das juntas.



Figura 1 - Robô Rhino XR4

As especificações básicas do robô Rhino XR4 estão resumidas na Tabela 1 (Rhino Robotics, 1997). Os acionamentos de junta são motores CC de ímã permanente com caixa de redução por engrenagens e codificadores ópticos incorporados. As juntas de base, ombro, cotovelo e cabeceio do pulso são acionadas por motores Pittman GM9413K046-R1; e a junta de rolagem do pulso e a abertura e fechamento da ferramenta, por motores Pittman GM8712E465. A transmissão mecânica dos motores para suas respectivas juntas é constituída por rodas dentadas e correntes metálicas.

Tabela 1 – Especificações do Robô Rhino XR4

Especificações	Valor	Unidades
----------------	-------	----------

Número de Eixos	5	
Capacidade de Carga	1,82	Kg
Máxima Velocidade da ponta	650	mm/s
Alcance Horizontal	628,6	mm
Alcance Vertical	889	mm
Peso	10,45	Kg
Repetibilidade	±1	mm

O sensoramento de posição e velocidade é realizado pelos codificadores óptico digitais montados nos eixos de alta velocidade dos motores. Os codificadores produzem os pulsos cujas transições de nível alto para nível baixo e vice-versa são detectadas. A contagem destas transições constituem as unidades do codificador (*contas*), a partir das quais determinam-se a posição e a velocidade da junta.

O controlador do robô Rhino XR4 é o Mark IV que não foi adquirido. O controlador Mark IV (Rhino Robotics, 1997) possui 8 canais de entrada/saída para o controle de até 8 eixos, e mais duas portas auxiliares. O Mark IV consiste de: a) um microprocessador de 16 bits, para controle geral; b) um microprocessador de 8 bits, para sensoramento e acionamento; e c) um microprocessador de 8 bits, para a unidade portátil de comando. A comunicação com o computador digital realiza-se através de uma interface RS-232C. A programação do controlador utiliza uma linguagem de comandos chamada *RoboTalk* através da qual controla-se o robô e a comunicação de entrada e saída.

3. CINEMÁTICA DO ROBÔ RHINO XR4

3.1 Cinemática direta

O problema da cinemática direta de um manipulador robótico é formulado como: Dado o vetor de variáveis de junta $\mathbf{q} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]^T$, determinar a posição \mathbf{p} e a orientação \mathbf{R} da ferramenta com referência a um sistema de coordenadas fixo na base do robô. Para resolver este problema, primeiro, aplica-se o algoritmo de Denavit-Hartenberg (SCHILLING, 1990) que aloca sistemas de coordenadas nos elos e determina os parâmetros cinemáticos $\{\theta_k, d_k, a_k, \alpha_k\}$. O diagrama de coordenadas nos elos e os parâmetros cinemáticos do robô Rhino XR4 são mostrados na Figura 2 e na Tabela 2 respectivamente. Segundo, determina-se a matriz do braço, ou seja, a matriz de transformação de coordenadas homogêneas composta T_{base}^{ferr} que mapeia as coordenadas da ferramenta nas coordenadas da base. Esta matriz é obtida multiplicando as matrizes de transformação homogênea adjacentes T_k^{k-1} . A matriz do braço resultante é:

$$T_{base}^{ferr} = \begin{bmatrix} R(q)_{3 \times 3} & p(q)_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1_{1 \times 1} \end{bmatrix} \quad (1)$$

onde:

$$R(q) = \begin{bmatrix} C_1 C_{234} C_5 + S_1 S_5 & -C_1 C_{234} S_5 + \\ S_1 C_{234} C_5 - C_1 S_5 & S_1 C_{234} C_5 + \\ -S_{234} C_5 & S_{234} S_5 \end{bmatrix} \quad p(q) = \begin{bmatrix} C_1 (a_2 C_2 + a_3 C_{23} + a_4 C_{234} - d_5 S_{234}) \\ S_1 (a_2 C_2 + a_3 C_{23} + a_4 C_{234} - d_5 S_{234}) \\ d_1 - a_2 S_2 - a_3 S_{23} - a_4 S_{234} - d_5 C_{234} \end{bmatrix}$$

sendo $C_i = \cos q_i$; $C_{ij} = \cos(q_i + q_j)$; $C_{ijk} = \cos(q_i + q_j + q_k)$

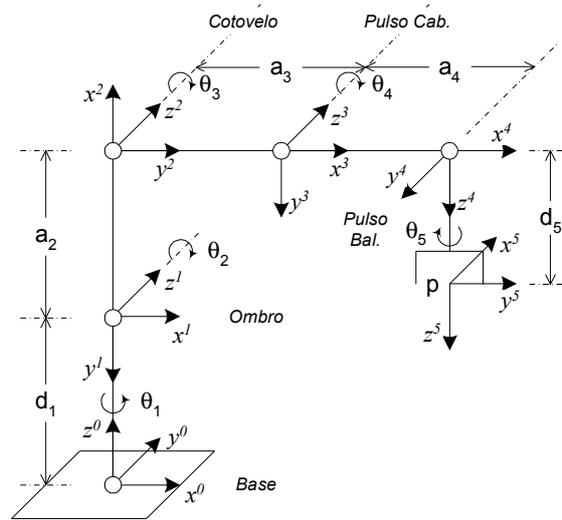


Figura 2 - Coordenadas no elos do robô Rhino XR4
Tabela 2 – Parâmetros Cinemáticos do robô Rhino XR4

Eixos	θ	d (mm)	a (mm)	α (graus)
1	θ_1	260,0	0	-90
2	θ_2	0	228,6	0
3	θ_3	0	228,6	0
4	θ_4	0	9,5	-90
5	θ_5	171,1	0	0

Em alguns robôs industriais, por razões de segurança ou em vista da aplicação, a transmissão mecânica acopla o movimento de algumas juntas. O robô Rhino XR4 também implementa este acoplamento por meio de um mecanismo de barras paralelas que acopla o movimento do ombro, cotovelo e o cabeceio do pulso. Como um exemplo, quando o motor do ombro é ativado para produzir uma rotação de Δq_2 , o cotovelo move-se simultaneamente produzindo uma rotação oposta de $\Delta q_3 = -\Delta q_2$. Como resultado o ângulo do cotovelo global Δq_{23} permanece constante. O mesmo acontece com o ângulo de cabeceio global Δq_{234} relativo aos incrementos de ângulo do ombro Δq_2 e do cotovelo Δq_3 . Como a análise cinemática prevista assume controle independente das juntas é necessário compensar este acoplamento. Isto é implementado no algoritmo de controle pré-processando o comando de movimentos de junta $\Delta \mathbf{q}_d$ desejado pela matriz de acoplamento \mathbf{C} para obter o comando $\Delta \mathbf{q}_n$ necessário.

$$\Delta \mathbf{q}_n = \mathbf{C} \cdot \Delta \mathbf{q}_d \quad (2)$$

O modelo cinemático do robô foi simulado no ambiente computacional Matlab pela função *rhcd* (\mathbf{q}) que retorna a matriz de transformação homogênea $T_{base}^{ferr}(\mathbf{q})$ (Eq. 1) para um valor específico do vetor de variáveis de junta \mathbf{q} . O desacoplamento (Eq. 2) é implementado pela função *rhcoup*(\mathbf{q}) produzindo o vetor de variáveis de junta necessário \mathbf{q}_n . Uma função adicional *rhmcld*(\mathbf{q}) implementa a cinemática direta de uma sequência de vetores \mathbf{q} útil no caso de geração de trajetórias.

3.2 Cinemática inversa

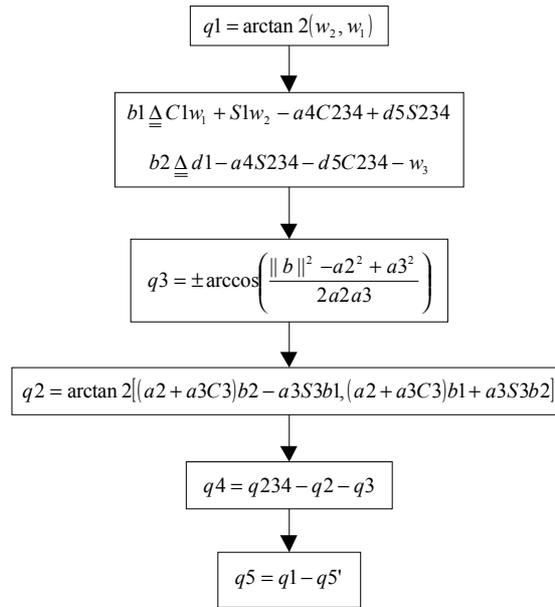


Figura 3 - Algoritmo de Cinemática Inversa para o robô Rhino XR4

O problema da cinemática inversa consiste em: Dada uma matriz de configuração da ferramenta, $\mathbf{T} = [\mathbf{p} \ \mathbf{R}]$, constituída pela posição \mathbf{p} e pela orientação \mathbf{R} , determinar os vetores de variáveis de junta \mathbf{q} que satisfazem a equação do braço (Eq. 1). Diferentemente à cinemática direta, onde existe uma solução geral, a solução analítica do problema inverso é específica de cada robô, assim, a existência e unicidade das soluções devem ser investigadas.

A carga computacional na solução de um problema de cinemática inversa pode ser reduzida utilizando uma caracterização alternativa da posição e da orientação desejadas da ferramenta. Aproveita-se o fato de haver informação redundante na matriz de rotação $\mathbf{R} = [\mathbf{r}^1 \ \mathbf{r}^2 \ \mathbf{r}^3]$, cujas colunas correspondem respectivamente aos vetores normal, de deslizamento e de aproximação da ferramenta. Uma alternativa (SCHILLING, 1990) é o vetor de configuração da ferramenta $\mathbf{w} \in \mathbb{R}^6$ definido como

$$\mathbf{w} = \begin{bmatrix} w_{1[3.x]} \\ w_{2[3.x]} \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ e^{(q_n/\pi)} \cdot \mathbf{r}^3 \end{bmatrix} \quad (3)$$

onde \mathbf{p} é o vetor de posição da ponta da ferramenta, sendo a orientação da ferramenta representada pelo vetor \mathbf{r}^3 , que indica os ângulos de cabeceio e de guinada da ferramenta. O fator de escala $\exp(q_n/\pi)$ representa o ângulo de rolagem do pulso q_n , sendo n a último eixo do manipulador. Uma simplificação é possível no caso do robô Rhino XR4 de 5 eixos, onde existe somente ângulo de cabeceio e de rolagem da ferramenta. Um vetor de configuração da ferramenta reduzido \mathbf{w}' pode ser definido para este robô

$$\mathbf{w}' = [p_1 \ p_2 \ p_3 \ q_{234} \ q_5']^T \quad (4)$$

onde: $q_{234} = q_2 + q_3 + q_4$; e $q_5' = q_1 - q_5$

q_{234} é o ângulo de cabeceio global (relativo à base) e q_5' o ângulo de rolagem global (relativo à base). Esta representação é mais eficiente e econômica, além de intuitiva, pois os ângulos q_{234} e q_5' são facilmente observáveis. A solução da cinemática inversa do robô Rhino XR4 foi implementada no ambiente Matlab pela função *rhei(q)*. Esta função implementa uma adaptação do algoritmo apresentado por SCHILLING (1990) e admite três tipos de dados de entrada: a) matriz de configuração da ferramenta (Eq. 1); b) vetor de configuração da

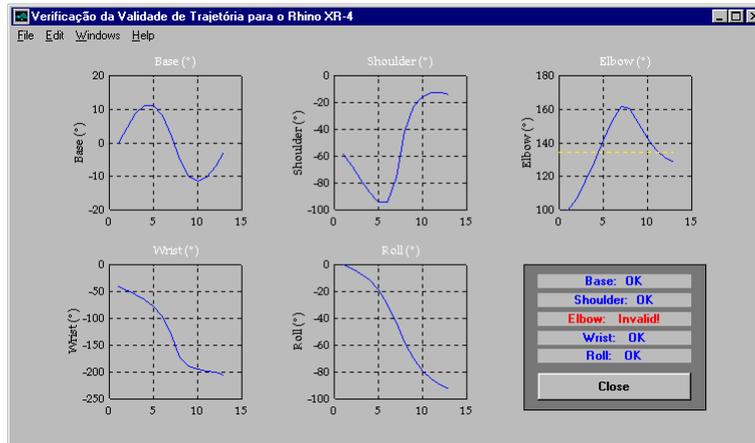


Figura 4 - Janela de verificação de validade de trajetórias ferramenta (Eq. 3); e c) vetor de configuração da ferramenta reduzido (Eq. 4). O algoritmo é mostrado na Figura 3.

4. SIMULAÇÃO GRÁFICA DA CINEMÁTICA DO ROBÔ RHINO XR4

Para testes de cinemática direta e inversa e para validação de trajetórias no planejamento das mesmas, integrou-se a simulação do modelo cinemático do robô (seção 3) a um programa de animação gráfica desenvolvido também no ambiente Matlab. Desenvolveu-se ainda uma interface de usuário usando as funções GUI (graphical user interface) do Matlab (MARCHAND, 1995). Os dados de entrada para o programa de animação são uma sequência de vetores de configuração da ferramenta w , resultado de uma amostragem espacial da trajetória desejada. Em seguida executando o algoritmo de cinemática inversa determinam-se os correspondentes vetores de variáveis de junta q . Aplicando esses vetores ao modelo de cinemática direta obtém-se a posição espacial (3D) da origem dos sistemas de coordenadas dos elos (L_n) (Fig. 2). Com esses pontos como referência, uma estrutura poligonal tridimensional do robô com arestas e vértices (*wire frame model*) é construída e representada na tela do monitor.

A simulação e animação gráfica são implementadas pela função *rhvis3(w)* que abre a janela de animação mostrada na Figura 4.

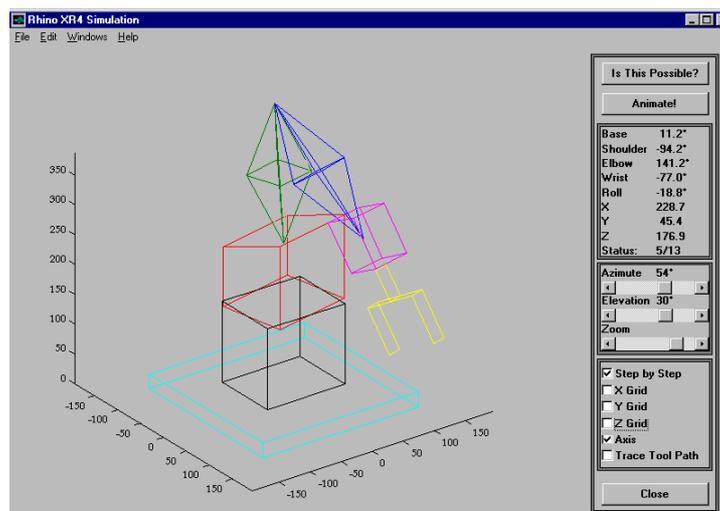


Figura 5 - Janela de Simulação do robô Rhino XR4

A janela apresenta a animação do robô e mostra dados e opções de simulação. Entre os dados mostrados estão: ângulos de junta q ; e posição da ponta da ferramenta p . Entre as opções de simulação tem-se: perspectiva de observação em termos dos ângulos de azimute e elevação; distância de observação (*zoom*); e movimento ponto a ponto da trajetória. A

animação acusa se a posição requerida da ponta da ferramenta passa para fora do volume de trabalho do robô. Uma função complementar chamada *rhvalid(w)* pode ser ativada na tela para verificar se os ângulos de junta requeridos estão dentro do volume de trabalho (Q) no espaço de variáveis de junta. A função mostra os gráficos dos ângulos de junta em função das amostras da trajetória, e indica os limites ultrapassados se isso acontecer. A janela aberta pela função é mostrada na Figura 5. A função *rhvalid* também pode ser executada independentemente da animação, desse modo é útil para validação de trajetórias.

5. CONTROLADOR MULTIAXIAL

Não tendo adquirido o controlador oferecido pelos fabricantes do robô Rhino XR4 e com informações sobre as limitações da sua programação (Mrad e Deeb, 1997) procedeu-se ao projeto de uma arquitetura de controle modular, expansível e de custo acessível. O diagrama funcional do controlador multiaxial construído é mostrado na Figura 6. Ele consiste de: a) um computador PC; b) uma placa de conversão de protocolo de comunicação serial Z232-485 (KERR, 1998); c) seis placas de controle de motores PIC-Servo (KERR, 1998); e d) fontes de alimentação de 24V para motores e de 7,5-9V para os circuitos lógicos. A modularidade foi adotada para futura expansão no controle de outros dispositivos e acessórios da estação robótica educacional.

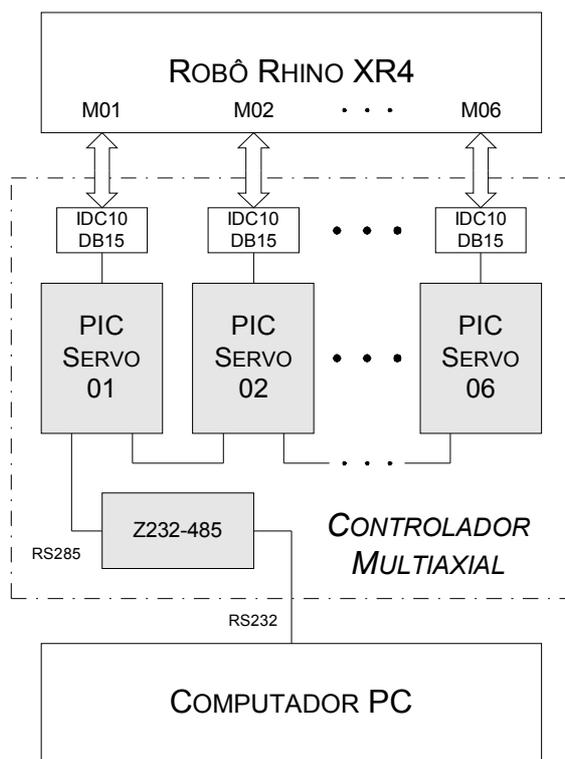


Figura 6 - Controlador Multi-axial

5.1 Placa de controle de motores

Várias alternativas para controle individual dos motores foram consideradas, incluindo ainda uma placa construída no laboratório. Finalmente, optou-se pela aquisição de um controlador integrado de movimento (*PIC-Servo Motor Control Board*) da J. R. Kerr Automation Engineering (KERR, 1998), cujo diagrama funcional é mostrado na Figura 7. Os principais componentes são:

PIC-SERVO: microcontrolador PIC16C73 programado com um algoritmo de controle PID e permitindo trajetórias nos perfis trapezoidal e de velocidade.

PIC-ENC: microcontrolador PIC16C54 programado como contador de 16 bits para os pulsos do codificador óptico digital.

LMD18201: circuito chaveador e amplificador de potência para saída PWM de até 3A contínuos.

LTC491: circuito de comunicação no protocolo RS485.

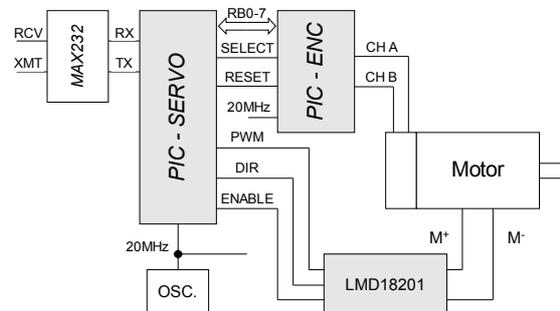


Figura 7 - Módulo PIC-Servo, para controle de motores

Tratando-se de um controlador integrado, este possui unidade de processamento e as seguintes interfaces: sensoreamento, acionamento e comunicação. A interface de sensoreamento é implementada pelo PIC-ENC que recebe dois canais (A e B) de pulsos em quadratura gerados pelo codificador óptico digital, detecta as transições havidas, incrementa ou decrementa seu contador de 16 bits e transfere a contagem para a unidade de processamento.

A unidade de processamento é implementada pelo PIC-SERVO que: a) determina a posição corrente do motor a partir dos dados transferidos pelo sensoreamento; b) implementa o perfil de movimento desejado a partir dos parâmetros de posição final, velocidade e aceleração; c) implementa uma ação de controle PID configurada pelos parâmetros de controle (ganhos PID; limites de integração, de saída e de corrente; e multiplicador da taxa de amostragem); e d) gera um sinal de controle de tipo PWM (onda quadrada de 20KHz modulada na largura dos pulsos). O movimento pode ser implementado em dois perfis: trapezoidal e de velocidade. Ambos os perfis operam calculando onde o motor deveria estar a cada instante de amostragem, gerando a posição desejada para o motor, aplicada então ao controlador PID para gerar o sinal PWM.

A interface de acionamento é implementada pelo acionador eletrônico LMD18201 que amplifica o sinal PWM, onde 0% de modulação corresponde a 0 e 100% a 255. A interface de comunicação é constituída pelo LTC491 que estabelece a transmissão de dados no protocolo RS485. Este protocolo de interface serial permite o controle de até 32 motores através da conexão em cadeia de 32 módulos PIC-Servo. Para conexão serial com o computador PC (protocolo RS232) utiliza-se o conversor de porta serial Z232-4845. A taxa de comunicação máxima é 115.200 bauds.

5.2 Cálculo e medida das resoluções de junta

Um parâmetro importante para efeitos de programação do controlador é a resolução angular dos sensores em unidades de graus/conta. Esta resolução foi primeiro calculada geometricamente a partir do eixo de alta velocidade do motor até a junta do eixo. Este cálculo leva em conta a redução de engrenagens do motor e a transmissão mecânica até as juntas.

Com a utilização do programa de diagnóstico *nettest* pode-se verificar a operação do controlador multiaxial no acionamento das juntas. As juntas do robô foram acionadas até seus limites e os ângulos varridos (*graus*) e a contagem de pulsos (*contas*) foram anotados. Assim, obteve-se a medida das resoluções das juntas em *graus/conta*, que foram comparadas com as

resoluções calculadas a partir da geometria da transmissão mecânica (engrenagens, rodas dentadas). As resoluções de junta calculadas e medidas são mostradas na Tabela 3.

Tabela 3 – Resoluções de junta do robô Rhino XR4

Junta	Calculada (graus/conta)	Medida (graus/conta)
Base	0,0572	0,0573
Ombro	0,0286	0,0288
Cotovelo	0,0286	0,0288
Cabeceio	0,0286	0,0288
Rolagem	0,0781	0,0782

5.3 Determinação do volume de trabalho no espaço de juntas

O volume de trabalho no espaço de juntas (Q) é o conjunto de todos os valores que as variáveis de junta (q) podem assumir. Como aplicação do controlador multiaxial determinou-se este volume levando as variáveis até seus limites. A Tabela 4 mostra os volumes de trabalho especificados pelo fabricante e o medido no laboratório. No experimento de medida limitou-se a excursão do ângulo da junta do ombro para 8,67 graus de modo a evitar a colisão do robô com a mesa.

Tabela 4 – Volume de trabalho no espaço de juntas

Junta	Especificada (graus)		Medida (graus)	
Base	-180	+180	-173,29	+174,20
Ombro	-135	+45	-135,89	+8,67
Cotovelo	-45	+135	-43,98	+134,29
Cabeceio	-225	+45	-224,59	+45,58
Rolagem	$-\infty$	$+\infty$	$-\infty$	$+\infty$

6. PROGRAMAÇÃO E INTERFACE DE USUÁRIO PARA CONTROLE DO ROBÔ

Após implementado o controlador multiaxial, a operação dos módulos de PIC-Servo foi verificada pelo uso de seus programas de diagnóstico *nettest* e *nmctest*, que testam as principais funções já implementadas nos módulos e a comunicação serial. Pode-se por exemplo alterar os parâmetros PID, mudar a taxa de comunicação (*baud rate*), executar o movimento do motor para uma posição ou velocidade pré-fixada, etc. Entretanto, estes programas não permitem o controle simultâneo de vários eixos e a execução de trajetórias mais complexas, entre outras limitações, assim, foi necessário desenvolver uma programação de controle do robô para execução de trajetórias e realização de tarefas. Optou-se pelo desenvolvimento no ambiente Borland C++ Builder de uma interface gráfica com múltiplas janelas para acesso às funções embutidas nos módulos PIC-Servo.

Nesta primeira versão a estratégia adotada tem duas etapas: Na primeira, os vetores de configuração da ferramenta (w) que compõem uma dada trajetória desejada são gerados matematicamente ou manualmente e salvos em um arquivo de dados em um formato previamente definido. A geração da trajetória no ambiente computacional Matlab é conveniente, pois pode-se verificar a validade da trajetória pelo programa *rhvalid(w)*, e também pode-se simular sua execução pelo programa *rhvis3(w)*.

Na segunda etapa, através da interface gráfica, as seguintes ações são realizadas: o arquivo de dados de entrada é selecionado; alguns parâmetros de execução são ajustados; as rotinas de cinemática inversa e desacoplamento de juntas são executadas; e o vetor de variáveis de junta (q) de referência é enviado aos módulos de controle. Para implementação desta etapa definiram-se três movimentos, a) Modo Posição, especificado pela posição desejada, velocidade máxima e aceleração, correspondendo ao perfil trapezoidal de velocidade; b) Modo Velocidade, especificado pela velocidade e aceleração, onde a velocidade

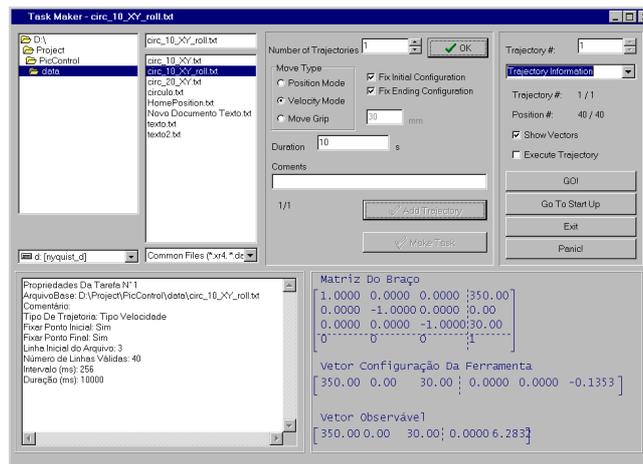


Figura 8 - Janela de Gerenciamento de Tarefas

do motor muda com a aceleração especificada até atingir a velocidade desejada; e c) Modo Ferramenta, que define a abertura ou fechamento da mesma. Na programação das variáveis posição, velocidade e aceleração, o deslocamento angular é medido em *contas* e o tempo em *marcações* do servo (múltiplos de 0,512ms).

Para a execução de tarefas implementou-se um programa que integra estes modos de movimento para todas as juntas do robô simultaneamente. São então declaradas algumas funções primitivas de movimento, que interagem diretamente com os módulos de controle (baixo nível) e funções de alto nível.

A interface gráfica é implementada pelo programa *PicControl.exe* e tem três menus principais:

PicServo - para configuração dos módulos de controle e ajuste dos parâmetros de operação. Possui os seguintes sub-menus: *Inicialização*, para escolha da porta e taxa de comunicação serial; *Parâmetros de Controle*, para escolha dos parâmetros PID, junta respectiva e limites de operação; *Habilitação de Módulos*, para seleção de eixo em operação; e *Configuração Padrão*, atalho para configuração com parâmetros de valor padrão.

Rhino XR4 - para acesso a funções específicas de configuração cinemática e controle das juntas do robô Rhino XR4. Possui um único sub-menu: *Início*, para ajustes cinemáticos e abre os menus *Janela*, *Inicialização* e *Calibração*. No primeiro, encontra-se o menu *Realização de Tarefas*, cuja janela é mostrada na Figura 8, que gerencia e monitora as tarefas. No segundo, comanda-se o robô para ir às posições de referência. No terceiro, executam-se procedimentos de calibração individual.

PicControl - para saída do programa e fechamento da porta serial.

7. CONCLUSÕES

O trabalho mostrou o estágio de desenvolvimento de uma estação robótica educacional já em operação. Um modelo cinemático para o robô foi formulado e verificado. Foram adquiridas somente a estrutura mecânica do robô e placas de controle individual de motores. Foram projetados e desenvolvidos: a) um ambiente de simulação da cinemática do robô com animação gráfica; b) um controlador multiaxial de arquitetura modular e inovadora; e c) a programação e a interface de usuário para controle do robô e execução de tarefas. A estação permitirá experiências de laboratório em cinemática, acompanhamento de trajetórias e execução de tarefas. Atividades já em andamento são: o desenvolvimento de uma interface de programação para controle da estação no ambiente Matlab; e a implementação de um sistema de visão por computador e a sua integração para guiagem do robô no espaço de trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

GUIMARÃES, H. A. **Estação Robótica Educacional: Interface de Programação e Desenvolvimento de Experiências**, 2002. Trabalho de Graduação, Departamento de Engenharia Elétrica, Universidade de Brasília.

KERR, J. R. - J. R. Kerr Automation Engineering. <http://www.jrkerr.com/>, 1998.

MRAD, F. T. and DEEB, G. Extending the Utility of the Rhino Educational Robot. **IEEE Transactions on Education**, EUA, 40(3): 184-189, 1997.

MARCHAND, P. **Graphics and GUI with Matlab**, CRC Press, FL, EUA, 1996.

Rhino Robotics (1997) - Catálogo de Produtos

RORIZ, V. **Estação Robótica Educacional baseada no Robô Rhino XR4: Controle de Juntas e Geração de Trajetórias**, 2000. Relatório Interno, Departamento de Engenharia Elétrica, Universidade de Brasília.

SCHILLING, R. **Fundamentals of Robotics: Analysis and Control**, Prentice-Hall, NJ, EUA, 1990.

DEVELOPMENT OF AN EDUCATIONAL ROBOTIC WORKSTATION

Abstract: *An educational robotic workstation is being developed around a Rhino XR4 robot, a five axis robot arm with electrical gripper, driven by 6 permanent DC servomotors with integral gearboxes and incremental encoders. A kinematic model for the robot was formulated and the direct and inverse kinematic problems were solved, for which Matlab programs were also written. For visualization purposes, a three-dimensional graphics simulator program was developed to simulate the robot kinematics by using a wire-frame model and the Matlab graphical user interface facilities. As only the robot mechanical structure was purchased, a multiaxial controller was designed and built using microcontroller-based motor control boards, a serial communication protocol converter and a PC computer. The controller was then used for checking the joint resolutions and the workspace geometry. The system programming was also developed, what includes an user interface, diagnosis, settings and robot control. The controller was then applied to follow trajectories and performing tasks.*

Keywords: *Control and robotics education, Robotics, Robot kinematics, Robot control, Robot simulation.*